TP1-RAID-LVM-STOCKAGE LOCAL - Sûreté de fonctionnement ECUE 3.2

samedi 22 mai 2021

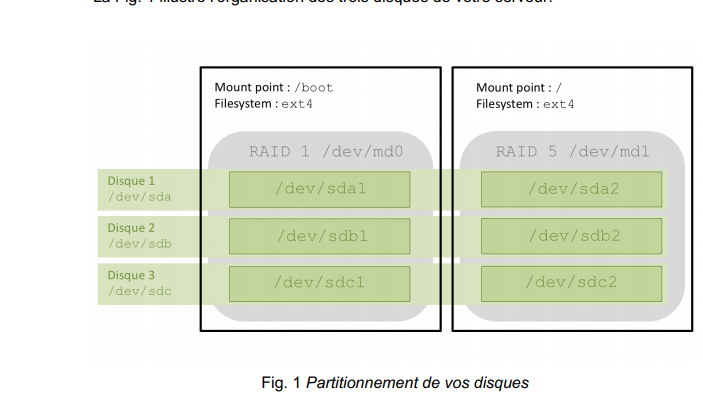
Amine ABDOUL-AZID

Kavirajan SARAVANANE

Martial SENE

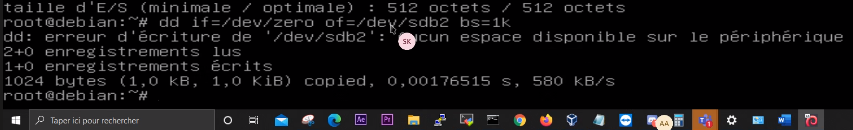
**PRÉPARATION :**

* RAID : Stockage redondant sur plusieurs disques d'un même hôte
* LVM : Agrégation et gestion flexible des volumes de stockage

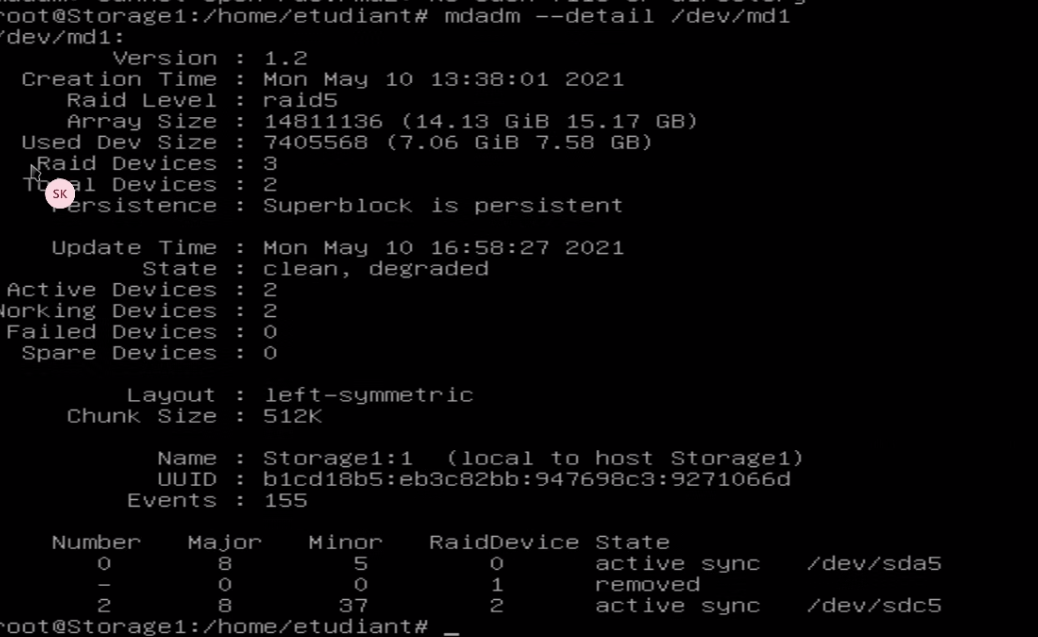


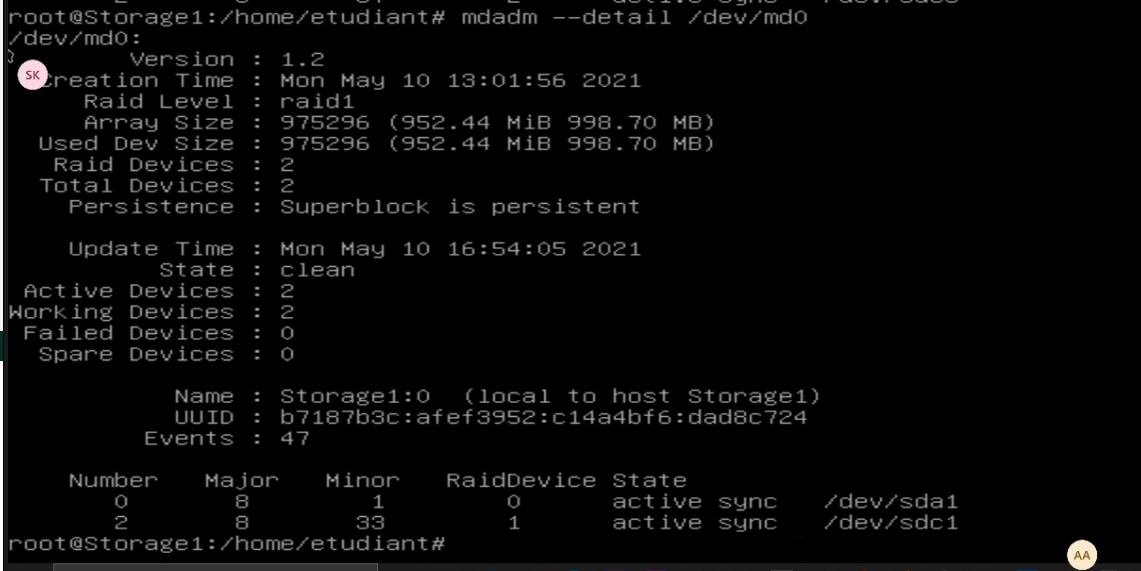
**Synthèse 1 :**

On efface en écrasant par des 0 le contenu de */dev/sdb2* avec une taille de bloc de 512 octets :

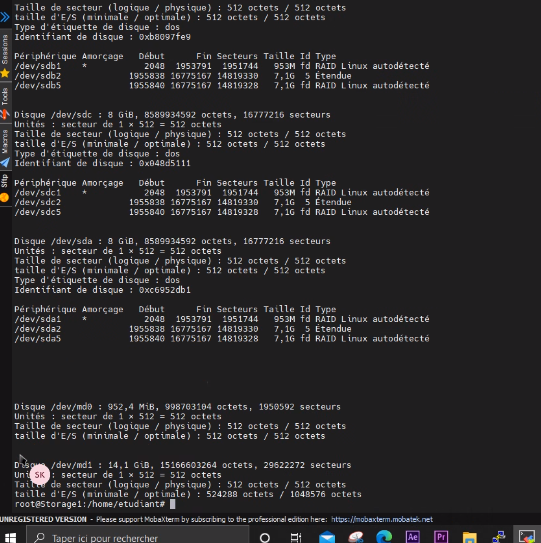


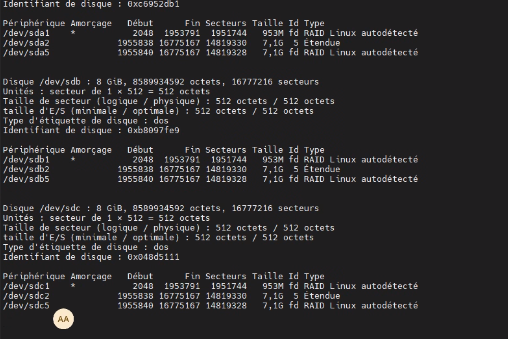
On utilise *mdadm* pour voir les détails du Volume RAID md1 :



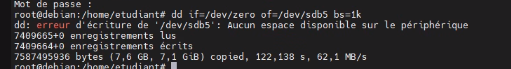
On utilise à nouveau la commande *mdadm* mais avec md2 :

On affiche les partitions avec *fdisk* :

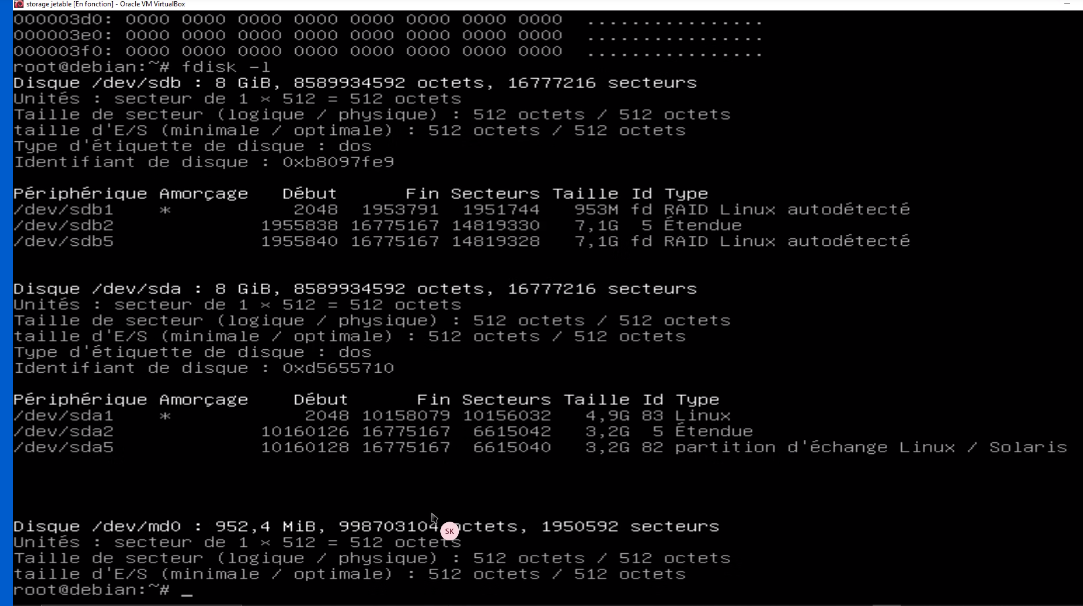




On a eu un bug avec dd qui refuse d’écrire sur la partition */dev/sdb5* :



Vérification avec *fdisk -l*



synth1:

on a réalisé l'installation, on a eu besoin de d'une image ISO de Debian

on a bien configuré. Pour cela, on a affiché leurs propriétés ainsi que

taille .

Storage1 et on a débranché son second disque. on a importé une VM Debian

comprendre pourquoi, on a inspecté ses logs et on a affiché les propriétés des volumes

ce disque au volume RAID 5.

l'état des volumes RAID et on a visualisé en continu sa reconstruction. Cette

à présent que vous souhaitiez , on a ajouté un espace de stockage

par exemple on a affiché la table des partitions de tous les disques .

on a créé de ce volume (en RAID 5), puis on a visualisé en continu sa

la configuration RAID persistante .

• on a formaté .

• on a monté dans l'arborescence globale.

lorsque vous on a démonté le volume.

**Synthèse 1 :**

Tout d’abord, on a configuré RAID et les différents volumes avec une image ISO de Debian.

Pour cela, on a affiché leurs propriétés avec *fdisk* et *mdadm* ainsi que leur taille avec *df*.

On a arrêté Storage1 et on a débranché son second disque. Puis on a importé une VM Debian jetable.

On a compris pourquoi Storage1 démarre avec un disque illisible : Il est sans partition MBR, donnée.

Ensuite on a ajouté ce disque au volume RAID 5. En affichant l'état des volumes RAID : On a visualisé en temps réel : Sa reconstruction. Ainsi RAID a « réparé le disque défectueux après analyse ».

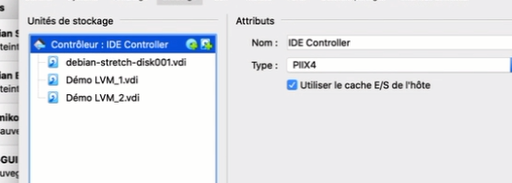
Désormais on a ajouté encore un espace de stockage à Storage1 et vérifier la table des partitions.

On a créé de ce volume RAID de niveau 5, puis on a visualisé en continu sa construction.

En outre, on a formaté ce volume et on a monté de façon persistante, démonté dans l'arborescence.

**Synthèse 2 :**

Les disques durs sont ajoutés de cette manière : *debian-stretch-disk001.vdi* en tant que disque dur primaire. Et le reste pour les volumes LVM.



On liste le dossier */dev*:

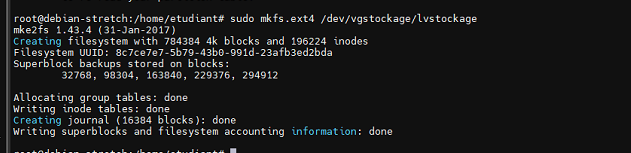


On repère le chemin et on teste les différents chemins pour formater :





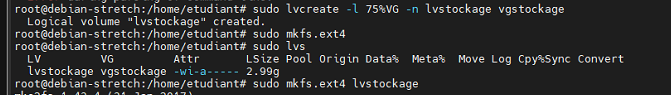
On formate en fonction du chemin */dev/vg\*/lv\** :



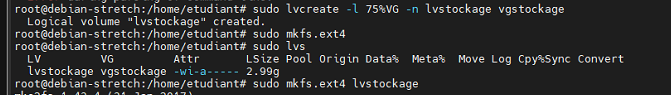
On créer les volumes :



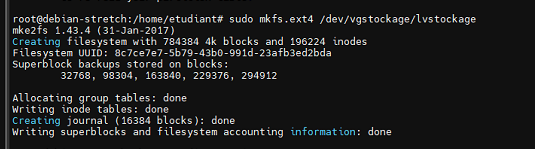
On crée le volume logique en fonction du groupe de volume en lui attribuant dans notre cas 75% du Volume logique :



On liste les volumes logiques en fonction de leur groupe de volume :



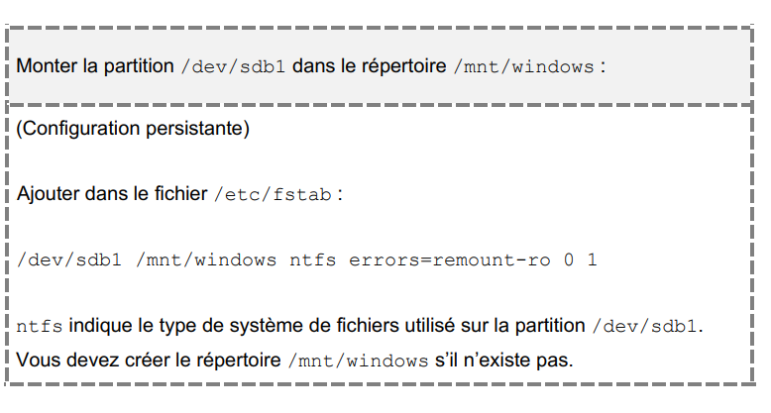
~~On formate~~



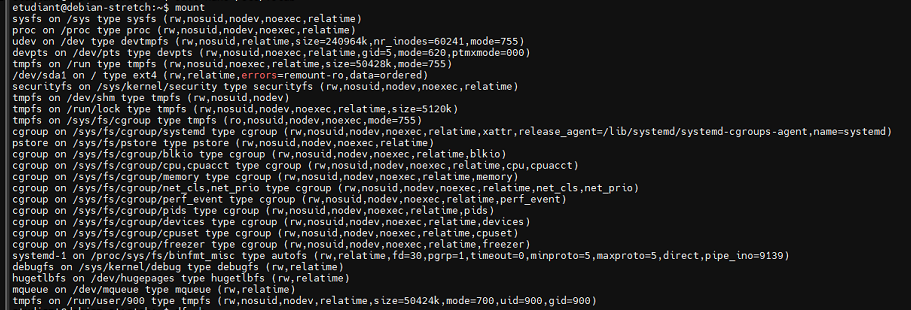
On a créé un point de montage et configuré *fstab* pour un montage permanent :



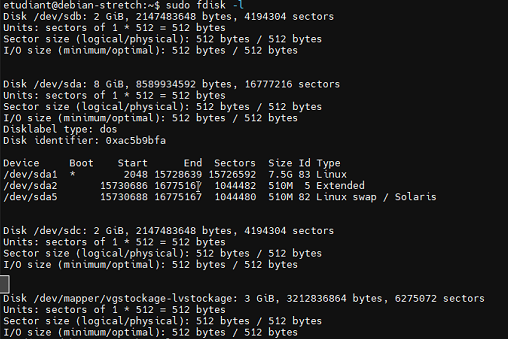
De cette manière :



On vérifie après redémarrage (ou du service *fstab* si cela fonctionne) avec la commande *mount* :

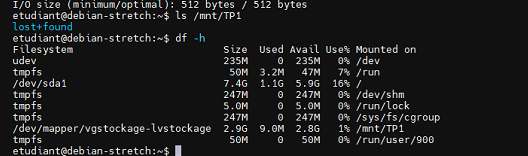


On liste les partitions avec *fdisk*

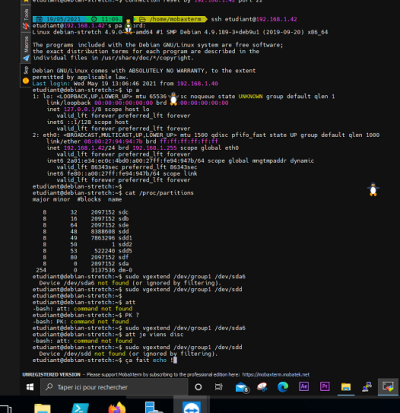


On liste le point de montage : Contient bien *lost+found*

On remarque la taille avec *df* :

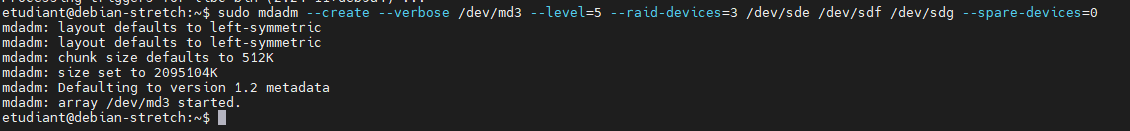


Usage de *vgextend* :



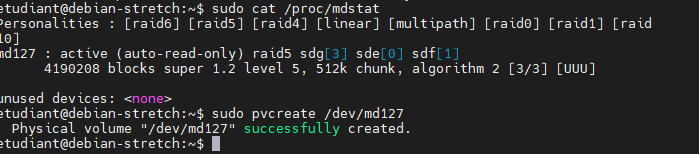
*Note : On a modifié notre chemin de montage /mnt/TP1 en tant que /mnt/windows suite à un changement de PC*

Après le montage des 3 disques SATA : On crée la partition */dev/md127* à partir de ces 3 disques

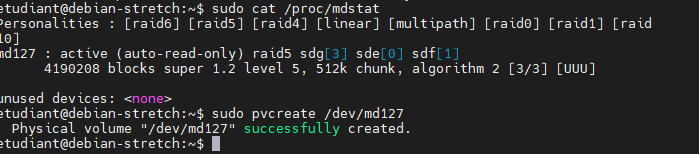


*Note : En fait on l’a nommé en tant que md127 :*

On liste les volumes RAID (md) actifs :



On initie ce volume RAID en tant que Physical Volume :



On liste pour vérifier la taille avant l’ajout dans vgstockage :

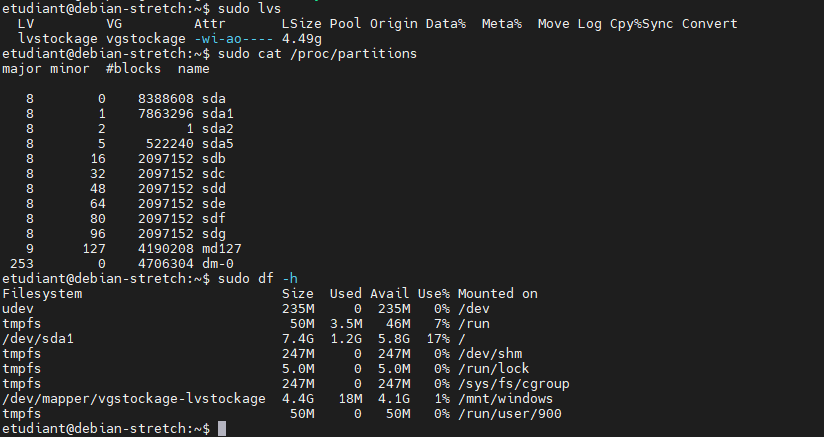


On se doit de mettre l’alias du vgstockage (=sans le chemin complet) et le chemin complet du volume RAID :





On peut vérifier que ce dernier est dans le VG vgstockage : Mais on n’a pas augmenté sa capacité :



On étend la capacité de disque avec 4 Go de + plus pour le lv lvstockage :



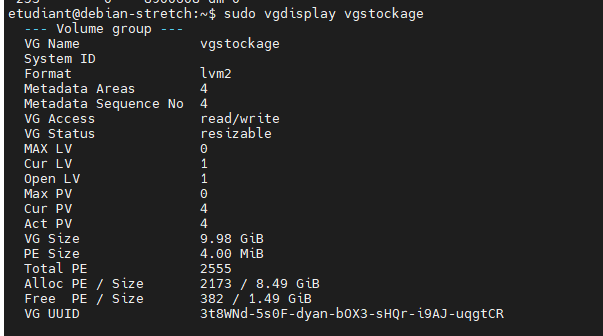
On remarque que le volume logique a été redimensionné : De 4.49Go à 8.49Go :

Mais que les partitions du système de fichiers ne détectent pas la modification :

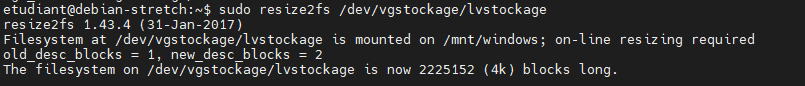


Par conséquent la taille du LV est supérieure (8.49GO) à la taille de la partition existant sur ce LV (4.49Go).

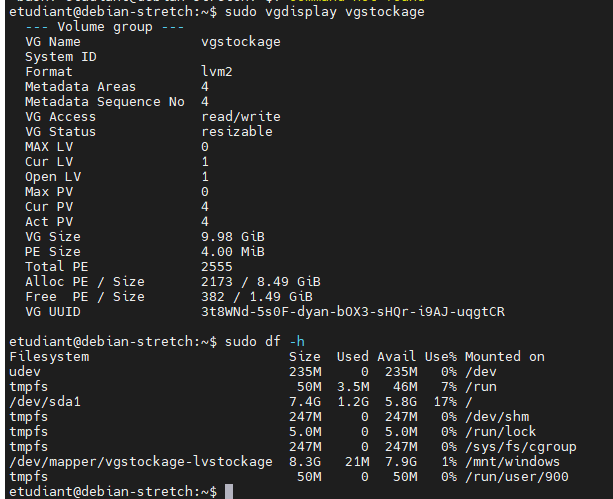
Et même le groupe de volume vgstockage affiche que la taille est de 8.49 et la taille maximale de 9.98



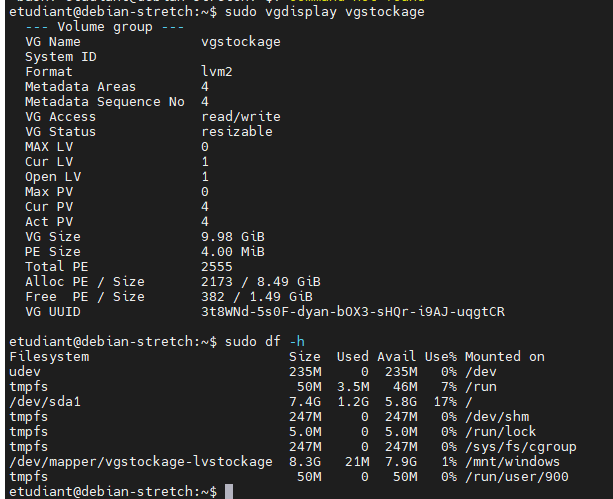
Pour remédier à cette situation, nous décidons d’étendre le système de fichiers avec *resize2fs sur le volume* logique:



On affiche à nouveau le vgstockage, on remarque que la taille est identique à la taille, avant d’étendre le système de fichiers :



Cependant comme on le prédisait désormais le système de fichiers détecte que la taille est de 8.49 Go (approximativement : 8.3Go)



synth2:

ensuite un VG on a nommé vgstockage. Pour l’instant notre VG est composé de

unique PV (vous , on a ajoutéez d'autres PV un peu plus tard) :

pouvez on a créé un premier LV on a nommé lvstockage et utilisant 75% de

• on a formaté en ext4

• on a monté dans l'arborescence globale. on a configuré un montage persistant .

cela, nous vous proposons d', on a ajouté un nouveau (et quatrième !) volume RAID

volume (en RAID 5).

ce volume en tant que PV , puis , on a ajouté ce dernier dans le VG

on a augmenté la capacité du LV : , on lui a ajouté 4 Go supplémentaires.

on ne le savait pas (encore). on a comparé la taille du LV et celle de la partition

on a utilisé la totalité de l’espace du LV, on a étendu son système de fichiers .

les propriétés du VG on a vérifié qu'il lui reste encore de l'espace non alloué.



Woo hoo ! Hasta la Vista ! On est génial n’est-ce pas !!

[**https://github.com/amineAUPEC/TP1-RAID-LVM\_LP.git**](https://github.com/amineAUPEC/TP1-RAID-LVM_LP.git)

**TP1 RAID LVM ECUE32 FIN :**